A Unified and Pipelined Hardware Architecture for Implementing Intra Prediction in HEVC

Yuebing Jiang, Daniel Llamocca, Marios Pattichis, and Gangadharan Esakki
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM 87106
Email: yuebing@ece.unm.edu, dllamocca@ieee.org, pattichis@ece.unm.edu, gesakki@ece.unm.edu

***Abstract*—The High Efficiency Video Coding (HEVC) standard can achieve significant improvements in coding performance over H.264/AVC. To achieve significant coding improvements in intra-predictive coding, HEVC relies on the use of an extended set of intra-prediction modes and prediction block sizes. This paper presents a unified hardware architecture for implementing all 35 intra-prediction modes that include the planar mode, the DC mode, and all angular modes for all prediction unit (PU) sizes ranging from 4×4 to 64×64 pixels. We propose the use of a unified reference sample indexing scheme that avoids the need for sample re-arrangement suggested in the HEVC reference design. The hardware architecture is implemented on a Xilinx Virtex 5 device (XC5VLX110T) for which we report power measurements, resource utilization, and the average number of required cycles per pixel.**

***Index Terms*—HEVC, Intra Prediction, Pipeline, FPGA, Hardware Architecture.**

I. Introduction

Among the most significant coding tools that improve over H.264/AVC, HEVC requires the implementation of extended prediction modes that need to work for different prediction unit (PU) sizes [1]. For intra-prediction, pre-decoded border pixels are used for predicting the entire PU. Depending on the prediction mode, predicted blocks can be reconstructed with: (i) the planar mode that uses bilinear interpolation. (ii) the DC mode that uses an average of the above and left reference pixels, (iii) the angular modes that use linear interpolation based on decoded pixels from the main-reference border or extrapolated reference pixels from the side-reference border. Under general testing conditions, PU sizes can vary from the smallest block of 4×4 to the largest block of 64×64. The goal of this paper is to provide a unifying framework that implements all of the 35 modes for block sizes from 4×4 to 64×64 while supporting parallelism in a pipelined architecture implementation.

To the best of our knowledge, current reported hardware implementations of the HEVC intra prediction either do not support all the prediction modes or don't support all the block sizes. In [2], the authors describe an intra-prediction architecture restricted to the angular modes on 4×4 blocks. The authors provide a *flexible reference sample selection* approach to minimize memory accesses. A more complete implementation was recently reported in [3]. In [3], the authors considered all except for the planar mode and gave implementations for all PU sizes. Instead of the unified approach proposed in this paper, the authors considered separate data paths for the horizontal and vertical data paths. To deliver higher performance, multipliers were implemented using custom shifters and adders that achieved a 500Mhz maximum operating frequency on IBM 65nm technology. In [4] and [5], the authors implemented the HEVC intra-prediction modes on 4×4 and 8×8 PU sizes using pixel equality based computation reduction (PECR) to reduce the energy consumption.

The motivation for our research comes from the need to provide an efficient implementation for all modes using a single circuit. The proposed approach uses unified reference sample indexing that avoids wasting cycles on re-arrangement or filling-in reference samples to a new buffer. A pipelined approach combines components from several modes to reduce the overall hardware footprint. Furthermore, the proposed pipelined circuit allows parallel processing of the different prediction modes that facilitates rate-distortion optimization that can be used to select an optimal mode for each block.

The rest of the paper is organized as follows. In section II, we describe the proposed unified approach. In section III, we discuss the pipelined implementation and system integration. We provide verification and implementation results on Virtex 5 FPGA in section IV. Concluding remarks are given in section V.

II. Unified Reference Sample Indexing and Accessing for HEVC Intra Prediction

HEVC intra prediction includes 33 angular direction modes (modes 2-34), an intra planar mode (mode 0), and an intra DC mode (mode 1)[6]. In the reference design provided by the standard draft and model software HM, intra prediction is implemented by re-arranging the reference samples into a new reference buffer according to the mode and then do the prediction. In the proposed approach, we will compute each prediction sample directly from original reference sample buffer using a unified indexing scheme and thus avoid the use of an intermediate buffer. The basic idea is that we can create a parallel and pipelined architecture for our unified indexing approach. Several modes were allowed in the pipeline to be computed at the same time for different stages without a lot of initial delay overhead.

A. Unified reference sample indexing

We setup the reference samples as shown in Fig. 1. For each (x, y) sample in the predicted block, we use a reference indexing equation (or equations) to map back to the reference sample(s) based on the prediction mode. For generality, we allow for different PU size ($nT \times nT$, nT=4,8,16,32,64), and map the *left/left-bottom* column and *upper/upper-right* row reference samples to the input buffer as shown in Fig. 1. To accommodate for the largest possible prediction unit, we allocate 257 bytes to the input buffer.

In what follows, let P_{y*nT+x} denotes the (x, y)-th prediction sample. We let R_i to index samples from the 1D input/reference buffer along the four sides and the upper-left corner pixel of the prediction block. Depending on the previous encoding/decoding conditions, the top-right row and left-bottom column pixels can be generated by extrapolation from the other two sides, and the $4nT+1$ reference pixels will always exist. Thus, our unified indexing approach computes P_{y*nT+x} using 1D indices from R_i.

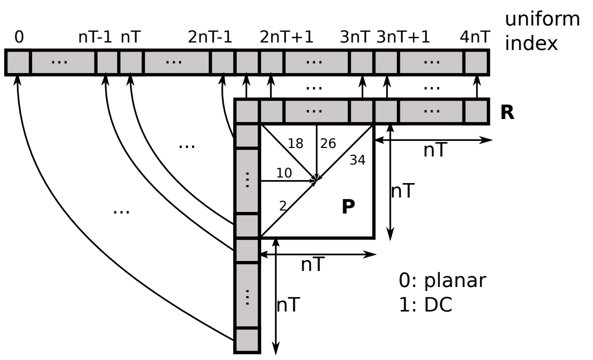


Fig. 1. Unified reference sample indexing for PU size of $nT \times nT$. Here, $4nT + 1$ reference pixels R are used to obtain $nT \times nT$ predicted samples P. The prediction directions are shown for several prediction modes (2,10,18,26,34).

B. Planar mode

In the planar mode, we use bilinear interpolation to generate P_{y*nT+x} using weighted-averaging over the 4 reference samples as given by:

$$\begin{aligned} P_{y*nT+x} \;=\; & ((nT-1-x)\cdot R_{2nT-1-y} \\ & +(x+1)\cdot R_{3nT+1} \\ & +(nT-1-y)\cdot R_{2nT+1+x} \\ & +(y+1)\cdot R_{nT-1}+nT) \\ & \gg (log2(nT)+1). \end{aligned} \tag{1}$$

where $x, y \in [0, nT-1]$ to cover all of the prediction block pixels.

C. DC mode

In the DC mode, the DC value is computed based on the average of all of the reference pixels except for corner pixels. Then, all of the internal pixels ($x, y \in [2, nT-1]$) are given by:

$$\begin{aligned} dcVal \;=\; & (\textstyle\sum_{x=0}^{nT-1} R_{2nT+1+x} + \sum_{y=0}^{nT-1} R_{2nT-1-y} \\ & +nT) \gg (log2(nT)+1) \\ P_{y*nT+x} \;=\; & dcVal. \end{aligned} \tag{2}$$

The first row and first column pixels use weighted averaging of the DC value and the nearest neighbour as given by:

$$\begin{aligned} P_{0*nT+0} &= (R_{2nT-1} + 2*dcVal + R_{2nT+1} + 2) \gg 2 \\ P_{0*nT+x} &= (R_{2nT+1+x} + 3*dcVal + 2) \gg 2 \\ & \quad x \in [1, nT-1] \\ P_{y*nT+0} &= (R_{2nT-1-y} + 3*dcVal + 2) \gg 2 \\ & \quad y \in [1, nT-1]. \end{aligned} \tag{3}$$

D. Angular mode

In the angular modes, the predicted pixels use at most two reference samples depending on the direction. Refer to Table I for the correspondence between the integer modes and the prediction angle parameters. A lookup table is used for computing the trigonometric parameters needed in the prediction. Directional modes are defined in terms of the prediction angle A_{mode}. Horizontal modes are defined for mode numbers $2-17$. For prediction, horizontal modes use **main reference** samples from the left column and **side reference** samples from the top row. Vertical modes are defined for mode numbers 18-34. Similar to horizontal modes, vertical modes use **main reference** samples from the top row and **side reference** samples from the left column. Extrapolation is needed when the prediction angle satisfies $tan(A_{mode}) < 0$ (modes $16-25$). The primary angular mode equation based on weighted average is given by:

$$\begin{aligned} P_{y*nT+x} &= ((32-w)*R_{tId0} + w*R_{tId1} + 16) \gg 5 \\ tId0 &= 2*nT + off0 \\ tId1 &= 2*nT + off1. \end{aligned} \tag{4}$$

To compute the weighting factor w, we use:

$$\begin{aligned} itm &= \begin{cases} (x+1)*T(mode) & \text{hor. mode} \\ (y+1)*T(mode) & \text{ver. mode} \end{cases} \\ c &= itm \gg 5 \\ w &= itm \;\&\&\; 31. \end{aligned} \tag{5}$$

For computing the indices $tId0$, $tId1$ of (4), we first compute the intermediate indices $index0, index1$ using:

$$\begin{aligned} index0 &= \begin{cases} y+c & \text{hor. mode} \\ x+c & \text{ver. mode} \end{cases} \\ index1 &= index0 + 1 \\ invA &= abs(AT(mode)). \end{aligned} \tag{6}$$

Then, we have the negative offsets given by:

$$\begin{aligned} neg\_off0 &= (128 + abs(index0+1)*invA) \gg 8 \\ neg\_off1 &= (128 + abs(index1+1)*invA) \gg 8, \end{aligned} \tag{7}$$

which can be used to compute $off0, off1$ using:

$$off0 = \begin{cases} neg\_off0; & index0 < \text{-1, hor. mode} \\ -1 - index0; & index0 \geq \text{-1, hor. mode} \\ -neg\_off0; & index0 < \text{-1, ver. mode} \\ 1 + index0; & index0 \geq \text{-1, ver. mode} \end{cases} \tag{8}$$

$$off1 = \begin{cases} neg\_off1; & index1 < \text{-1, hor. mode} \\ -1 - index1; & index1 \geq \text{-1, hor. mode} \\ -neg\_off1; & index1 < \text{-1, ver. mode} \\ 1 + index1; & index1 \geq \text{-1, ver. mode.} \end{cases} \tag{9}$$

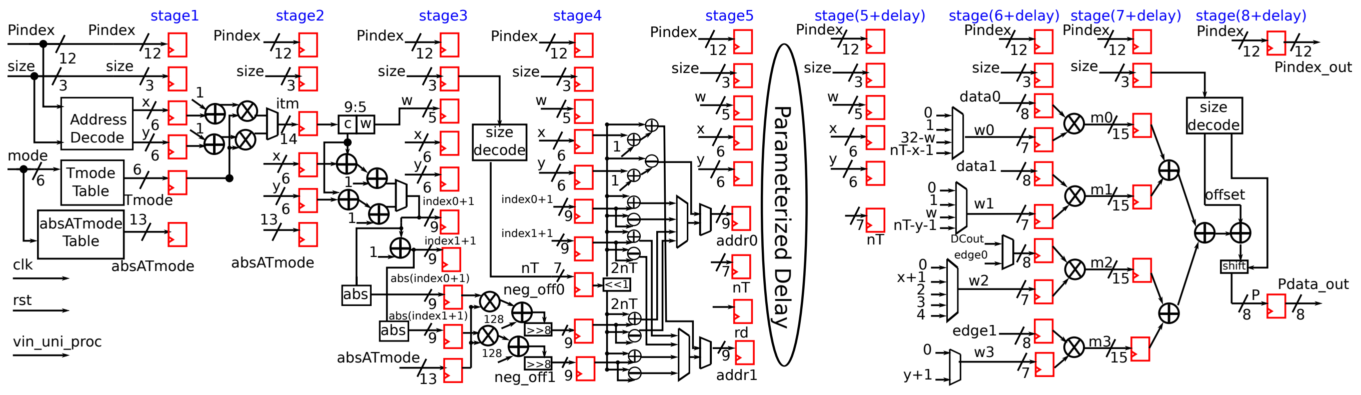


Fig. 2. Datapath for the pipelined *uni_proc* circuit using $size = log2(nT) - 2$, $Pindex = y \times nT + x$, and $mode \in [0, 34]$. Parameter *delay* is used to notify number of cycles for RAM operation between address assertion and data to be ready. When using BRAM on virtex 5 FPGA, $delay = 2$.

TABLE I
LOOK UP TABLE FOR ANGLE A_{mode} PARAMETERS FOR ANGULAR PREDICTION MODE. $T(mode) = 32 \times tan(A_{mode})$, $AT(mode) = 256 \times actan(A_{mode})$

mode	2	3	4	5	6	7	8
$T(mode)$	32	26	21	17	13	9	5
$AT(mode)$	—	—	—	—	—	—	—
mode	9	10	11	12	13	14	15
$T(mode)$	2	0	-2	-5	-9	-13	-17
$AT(mode)$	—	—	-4096	-1638	-910	-630	-482
mode	16	17	18	19	20	21	22
$T(mode)$	-21	-26	-32	-26	-21	-17	-13
$AT(mode)$	-390	-315	-256	-315	-390	-482	-630
mode	23	24	25	26	27	28	29
$T(mode)$	-9	-5	-2	0	2	5	9
$AT(mode)$	-910	-1638	-4096	—	—	—	—
mode	30	31	32	33	34		
$T(mode)$	13	17	21	26	32		
$AT(mode)$	—	—	—	—	—		

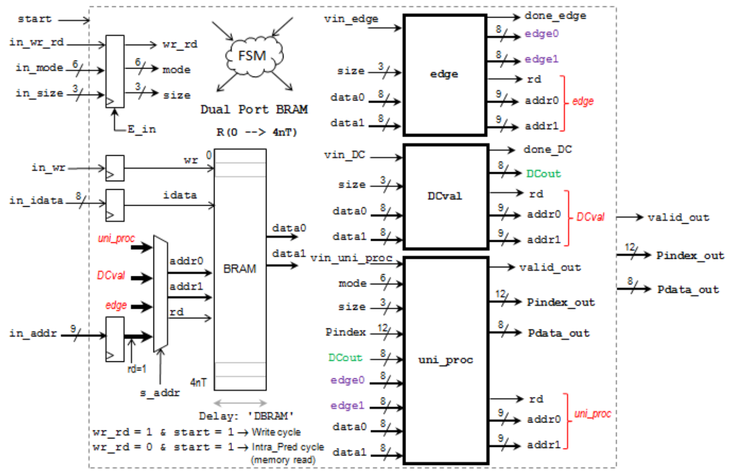


Fig. 3. Integrated system integration using pipelined *uni_proc* circuit.

III. METHODOLOGY AND IMPLEMENTATION

A. Pipeline Organization

We unify all of the modes using a pipelined approach. The number of stages of the pipeline will depend on the RAM type used to save the reference pixels. We provide the pipelined architecture in Fig.2 and summarize the pipeline stages below:

Stage 1: Compute $T(mode)$ and $AT(mode)$ using the lookup table. Determine x and y from 1D index P_{index}.

Stage 2: Compute itm as given in eq.5.

Stage 3: Compute c and w as given in eq.5.
Generate $index0$ and $index1$ as given in eq.6.

Stage 4: Compute neg_off0 and neg_off1 as given in eq.7.

Stage 5: Compute $addr0 = tId0$ and $addr1 = tId1$ as given in eq. 4.
In angular mode, compute $off0$ and $off1$ as given by eq. 8 and 9.
In DC or planar mode compute:
$off0 = -1 - y$ and $off1 = 1 + x$.

Stage(s) 6 : 5 + delay: Based on RAM type, we need:
One cycle (delay=1) for register-based RAM or
Two cycle delay (delay=2) for dual-port BRAM.

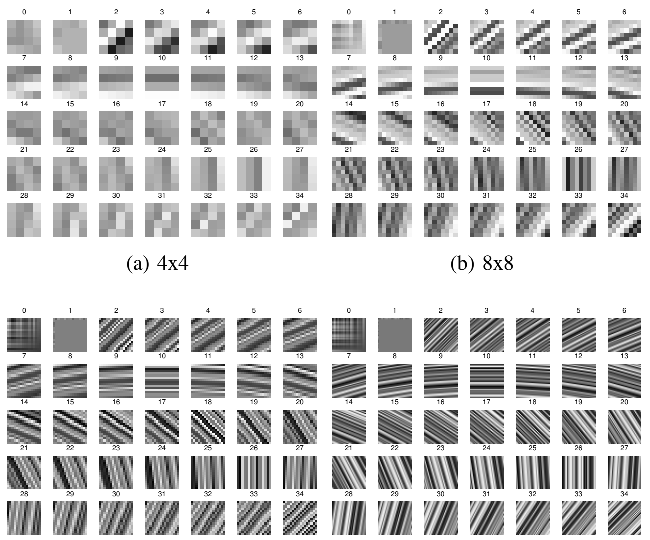
Stage 6 + delay: Use $data0 = R_{addr0}$ and $data1 = R_{addr1}$. in the following computations.
For Planar or DC mode:
Select $edge0$, $edge1$ or $dcVal$ based on the mode.
Select the weights for input pixels $w0, w1, w2, w3$.

Stage 7 + delay: Compute $m(i) = w(i) \times R_{tId(i)}, i \in [0-3]$.

Stage 8 + delay: Compute predicted pixel P_{data_out} as:
$P_{data\_out} = (m0 + m1 + m2 + m3 + offset) \gg shift$.

B. System Integration

The integrated system is given in Fig. 3. The system consists of 4 components *BRAM*, *edge*, *DCval* and *uni_proc* circuit. The *BRAM* circuit is used to store the decoded reference samples. The *edge* circuit is only used in planar mode for accessing R_{nT-1} and R_{3nT+1}. The *DCval* circuit is only used in the DC mode for computing $dcVal$. The *uni_proc* circuit implements the pipeline stages described in subsection III-A. The integrated system uses a finite state machine to control the 4 component circuits.



(a) 4x4 (b) 8x8

(c) 16x16 (d) 32x32

Fig. 4. 35 modes of intra prediciton for HEVC, for PU sizes from 4x4 to 32x32, with random generated reference samples for each PU size.

IV. Results

A. Unified Reference Sample Indexing Verification

To verify the correctness of our unified indexing and accessing method, we generated the predicted blocks for each PU size $nT = 4, 8, 16, 32, 64$ for randomly generated reference samples. For each PU size and set of random reference samples, we confirm that the predicted Blocks were identical to the ones produced by the reference software.

We provide examples for PU sizes from 4×4 to 32×32 in Fig.4. From the results, we can see the fine directional selectivity of the HEVC standard.

B. Synthesis Results

The proposed architecture was coded in VHDL and synthesized using Xilinx ISE 13.2 speed grade -3 for the $xc5vlx110t$ device. Synthesis results are shown in Table II.

TABLE II
SYNTHESIS RESULTS ON XC5VLX110T WITH SPEED GRADE -3 USING XILINX ISE 13.2

	uni_proc	*whole system*	FPGA
# of slice registers	562	684	69120
# of slice LUTs	467	690	69120
# of DSP48Es	7	7	64
# of BRAM	0	1	148

From the synthesis report, we have the maximum delay path is 4.887ns that gives a maximum operating frequency is 204.61Mhz. For the planar mode, we have a setup latency of $delay$ cycles (defined in section III-A) for the *edge* circuit. DC mode has setup latency of $2nT - 1 + delay$ cycles due to the *DCval* circuit. All other modes have no additional setup delays. After setup, the pipeline in uni_proc is filled in and after $8 + delay$ cycles, the pipelined is filled up and the first output pixel is computed. For the remaining pixels, we need an additional $nT \times nT - 1$ cycles since we output one pixel per cycle. For the encoder to determine the best mode, it generates all 35 modes prediction results. Overall, it takes $3delay + 7 + 2nT + 35nT^2$ cycles to generate all $35 \cdot nT^2$ output pixels. Thus, on average, the setup delay is dominated by the number of output pixels $35 \cdot nT^2$. Based on the encoded mode, the decoder circuit will only need $setup\_latency + 8 + delay + nT^2$ cycles to compute all nT^2 output pixels. For each mode and PU size, we summarize the operation cycles in Table III. At an operating frequency of 100Mhz, the fully realized system is simulated after place and route, and gives a dynamic power cost of 50.48mW.

We also provide comparisons of our approach to previously published results. Compared to [2], we provide more modes, PU sizes and higher operating frequency (204Mhz vs 150Mhz). In terms of resource usage, we are using less resources in Virtex 5 than the Virtex 6 resources reported in [4] and [5].

TABLE III
TOTAL CYCLES TO GENERATE ONE PREDICTION BLOCK AND AVERAGE CYCLES FOR ONE PREDICTION PIXEL, ON DECODER SIDE, DELAY=2.

	DC		Plannar		Angular	
	total	avg.	total	avg.	total	avg.
4x4	35	2.19	28	1.75	26	1.63
8x8	91	1.42	76	1.19	74	1.16
16x16	299	1.17	268	1.05	266	1.04
32x32	1099	1.07	1036	1.01	1034	1.01
64x64	4235	1.03	4108	1.00	4106	1.00

V. CONCLUSION

In this paper, we presented a pipelined and unified hardware architecture for intra prediction in HEVC. The proposed approach provides a pipelined architecture that can compute all 35 modes for block sizes from 4×4 to 64×64 using a single integrated system. We also demonstrate the efficiency of the approach by reporting dynamic power consumption, average number of cycles per pixel, and required hardware resources.

References

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Trans. On Circuits And Systems For Video Technology*, p. to be published, 2012.

[2] F. Li, G. Shi, and F. Wu, "An efficient vlsi architecture for 4x4 intra prediction in the high efficiency video coding (hevc) standard," in *2011 18th IEEE International Conference on Image Processing (ICIP),*, 2011, pp. 373–376.

[3] D. Palomino, F. Sampaio, L. Agostini, S. Bampi, and A. Susin, "A memory aware and multiplierless vlsi architecture for the complete intra prediction of the hevc emerging standard," in *2012 19th IEEE International Conference on Image Processing (ICIP),*, 2012, pp. 201–204.

[4] E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A high performance and low energy intra prediction hardware for high efficiency video coding," in *2012 22nd International Conference on Field Programmable Logic and Applications (FPL),*, 2012, pp. 719–722.

[5] ——, "A high performance and low energy intra prediction hardware for hevc video decoding," in *2012 Conference on Design and Architectures for Signal and Image Processing (DASIP),*, 2012, pp. 1–8.

[6] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, Y.-K. Wang, and T. Wiegand, "High efficiency video coding (hevc) text specification draft 10," in *JCTVC-L1003*, Jan. 2013.